REMARKS

This application has been carefully reviewed in light of the Office Action dated May 20, 2004. Reconsideration and favorable action in this case are respectfully requested.

Applicants note with appreciation that the Examiner has allowed claim 8-11 and has indicated that claims 3-5 and 12-16 are allowable if rewritten in independent form.

The Examiner has rejected claims 1 and 7 under 35 U.S.C. §102(b) as being unpatentable over U.S. Pat. NO. 4,642,756 to Sherrod. Applicants have reviewed this reference in detail and do not believe that it discloses or makes obvious the invention as claimed.

The Examiner has rejected claims 1-2 and 6-7 under 35 U.S.C. §103(a) as being unpatentable over U.S. Pat. No. 6,430,640 to Lim, in view of U.S. Pat. No. 5,588,111 to Cutts, Jr. et al (hereinafter "Cutts"). Applicants have reviewed these references in detail and do not believe that they disclose or make obvious the invention as claimed.

With regard to the Sherrod reference, Applicants note that the reference is not directed towards a plurality of devices initiating access requests to a shared resource. Sherrod is directed towards a *single device* (CPU 5') executing a plurality of *tasks*. In this regard, Sherrod teaches a task scheduler 6' which handles the scheduling of tasks (col. 3, line 63 through col. 4 line 29). Each *task* (not access request) includes two priority values – an internal priority provided by the task scheduler 6' and an external priority assigned by the computer operator or the task itself. The internal priority may change during execution of the task, while the external priority remains fixed (col. 4, lines 30-43).

The internal and external priorities are used by the task scheduler 6' to order current tasks in a list. The list is arranged by internal priority; if two tasks have the same

internal priority, the task with the higher external priority is placed higher to the top of the list (col. 5. lines 35-53).

The invention defined by claim 1 is directed to method for prioritizing access to a shared resource in a digital system having a plurality of devices vying for access to the shared resource. Access requests are initiated by each of the plurality of devices, where each access request has two priority values. Access to the shared device is arbitrated by the shared device by using the higher of the two priority values.

In Sherrod, there are no shared resources. The CPU 5' is in charge of all access requests to the RAM 1' and ROM 2'. The task scheduler 6' does not access the ROM 1' or RAM 2'.

Even if the CPU 5' was considered the shared resource, there are no access requests sent from the task scheduler 6' to the CPU 5' – decisions on which task to run are handled internally by the task scheduler. Further, the method used for arbitrating between tasks in the Sherrod is different than that specified in claim 1. In Sherrod, the tasks are prioritized by internal priority value; the external priority value only breaks a tie between tasks with the same internal priority value.

For the same reasons as stated above, claim 7 is not anticipated by Sherrod.

In the previous responses, Applicants have asserted that, Lim does not teach or even suggest the novel arbitration method and apparatus claimed in Applicant's base Claims 1 and 7. It appears that the Examiner is in agreement that Lim merely teaches the dynamic assigning of a single priority value to each processor in a multiprocessor system. (Lim: Col 15, lines 9-29). The Examiner contends, however, that Cutts teaches the use of two priority values, namely the PID and VPN values.

Applicant respectfully disagrees with this analysis. Both the PID and VPN are parts of a logical address submitted to the TLB for translation to a physical address. The

PID is a process identifier which is an extension of the virtual address to form a unique address (column 28, lines 13-18). The PID is used by the TLB to distinguish between different processes operating in the same logical address range. Cutts does not identify any process in which the PID is used for priority to access a resource.

The PID value is mentioned in only two paragraphs in the Cutts disclosure, set forth below:

> In FIG. 19, two separate 2GByte virtual address spaces 150 and 151 are illustrated; the processor 40 operates in one of two modes, user mode and kernel mode. The processor can only access the area 150 in the user mode, or can access both the areas 150 and 151 in the kernel mode. The kernel mode is analogous to the supervisory mode provided in many machines. The processor 40 is configured to operate normally in the user mode until an exception is detected forcing it into the kernel mode, where it remains until a restore from exception (RFE) instruction is executed. The manner in which the memory addresses are translated or mapped depends upon the operating mode of the microprocessor, which is defined by a bit in a status register. When in the user mode, a single, uniform virtual address space 150 referred to as "kuseg" of 2-GByte size is available. Each virtual address is also extended with a 6-bit process identifier (PID) field to form unique virtual addresses for up to sixty-four user processes. All references to this segment 150 in user mode are mapped through the TLB 83, and use of the caches 144 and 145 is determined by bit settings for each page entry in the TLB entries; i.e., some pages may be cachable and some not as specified by the programmer. (paragraph starting at col. 27, line 66)
>
> The 32-bit virtual addresses generated in the registers 76 or PC 80 of the microprocessor chip and output on the bus 84 are represented in FIG. 20, where it is seen that bits 0-11 are the offset used unconditionally as the low-order 12-bits of the address on bus 42 of FIG. 3, while bits 12-31 are the VPN or virtual page number in which bits 29-31 select between kuseg, kseg0, kseg1 and kseg2. The process identifier PID for the currently-executing process is stored in a register also accessible by the TLB. The 64-bit TLB entries are represented in FIG. 20 as well, where it is seen that the 20-bit VPN from the virtual address is compared to the 20-bit VPN field located in bits 44-63 of the 64-bit

> entry, while at the same time the PID is compared to bits 38-43; if a match is found in any of the sixty-four 64-bit TLB entries, the page frame number PFN at bits 12-31 of the matched entry is used as the output via busses 82 and 42 of FIG. 3 (assuming other criteria are met). Other one-bit values in a TLB entry include N, D, V and G. N is the non-cachable indicator, and if set the page is non-cachable and the processor directly accesses local memory or global memory instead of first accessing the cache 44 or 45. D is a write-protect bit, and if set means that the location is "dirty" and therefore writable, but if zero a write operation causes a trap. The V bit means valid if set, and allows the. TLB entries to be cleared by merely resetting the valid bits; this V bit is used in the page-swapping arrangement of this system to indicate whether a page is in local or global memory. The G bit is to allow global accesses which ignore the PID match requirement for a valid TLB translation; in kseg2 this allows the kernel to access all mapped data without regard for PID. (paragraph starting at col. 25, line 53)

These paragraphs clearly define the PID as merely a process identifier, not a priority value.

The VPN is the virtual page number, comprising bits 12-31 of a virtual address. The VPN points to a page of memory space. The VPN is translated by the TLB to a physical page address that is concatenated with the lower order bits of the virtual address to form an address in the physical memory (column 28, line 53 through column 29, line 16). Once again, there is no indication in Cutts that the VPN has any function in prioritizing requests to access any resource.

Applicants sincerely believe that neither the VPN nor the PID have any purpose in prioritizing memory access requests. They are simply fields which allow the proper physical address to be determined from a logical address. *If the Examiner disagrees, however, Applicants would greatly appreciate a more detailed explanation from the Examiner on how these fields could be used to prioritize between requests from different devices.* Applicants do not believe that it is possible that any system would rely on a

process identifier and a virtual page number to decide which access request should be serviced first, since neither of the fields is related to the necessity of greater access speed.

An extension of one month is requested and a Request for Extension of Time under § 1.136 with the appropriate fee is attached hereto.

The Commissioner is hereby authorized to charge any fees or credit any overpayment, including extension fees, to Deposit Account No. 20-0668 of Texas Instruments Incorporated.

Applicants have made a diligent effort to place the claims in condition for allowance. However, should there remain unresolved issues that require adverse action, it is respectfully requested that the Examiner telephone Alan W. Lintel, Applicants' Attorney at (972) 664-9595 so that such issues may be resolved as expeditiously as possible.

For these reasons, and in view of the above amendments, this application is now considered to be in condition for allowance and such action is earnestly solicited.

Respectfully Submitted,

Alan W. Lintel
Attorney for Applicant(s)
Reg. No. 32478

September 20, 2004
Anderson, Levine & Lintel
14785 Preston Rd.
Suite 650
Dallas, Texas 75254
Tel. (972) 664-9595